

## SERIE D'APPLICATIONS

Application 1 :

Ecrire un algorithme et un script Python intitulé « inverse » qui permet de saisir une chaîne de caractères de longueur paire et d'afficher cette chaîne inversée.

Exp : si ch= " info" alors le programme affichera "ofni"

Application 2 :

Ecrire un algorithme et un script Python intitulé « FACTORIEL », qui permet de calculer et d'afficher le factoriel d'un entier N positif et < 10.

Exp : si n= 6 alors le programme affichera :  $4! = 1*2*3*4 = 24$

Application 3 :

Ecrire un algorithme intitulé **SOMME** qui permet de calculer et d'afficher la somme « s » d'un ensemble de valeurs « n » à saisir sachant que cette liste se termine par -1 ( et que -1 ne fait pas partie de la somme).

Exp : 5, -3, 2, -1 => s = 4

Application 4 :

Ecrire un algorithme « **PRODUIT** » et un **script Python** qui permet de calculer et d'afficher le produit « P » d'une liste d'entiers saisis « x ». La liste se termine par 0. (Sachant que 0 ne fait pas partie du produit).

Exp : 3, 2, -7, 0 => alors le programme affichera le produit = -70

Application 5 :

Ecrire un algorithme et un script Python qui permet de calculer et d'afficher le PGCD de deux entiers A et B saisis **strictement positifs** selon le principe suivant :

Tant que A ≠ B  
 Si A > B alors  
 A ← A - B  
 Sinon  
 B ← B - A

Application 6 :

Ecrire un algorithme intitulé « DIVISEUR » qui permet de saisir un entier « n » strictement positif, de déterminer et d'afficher tous ses diviseurs.

Exp : si n = 12 alors le programme affichera : les diviseurs : 1, 2, 3, 4, 6, 12

Application 7 :

Un nombre parfait est un entier positif caractérisé par le fait qu'il est égal à la somme de tous ses diviseurs sauf lui-même.

Le premier nombre parfait est  $6 = 1 + 2 + 3$  qui sont les diviseurs de 6.

Ecrire un algorithme intitulé « PARFAIT », qui permet de saisir un entier « n » > 1 et de vérifier s'il est parfait ou non.



في دارك... إتهون علمي قرابتة إصغارك

